JW AF

Our Docket No.: 1496.00047

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re Application of:

Applicant	Ariel Cohen et al.		
Application No.:	09/746,796	Examiner:	Harkness, C.
Filed:	December 22, 2000	Art Group:	2183
For:	MICROCODE BASED HARDWARE TRANSLATOR TO SUPPORT A MULTITUDE OF PROCESSORS		

CERTIFICATE OF MAILING

I hereby certify that this letter, the response or amendment attached hereto are being deposited with the United States Postal Service as first class mail in an envelope addressed to Commissioner for Patents, P.O. Box 1450, Alexandria, VA 22313-1450, on January 6, 2005.

By: Jan M. Dunbar
Jan M. Dunbar

RESPONSE TO NOTIFICATION OF NON-COMPLIANT APPEAL BRIEF (37 C.F.R. §41.37)

Commissioner For Patents
P.O. Box 1450
Alexandria, VA 22313-1450

Sir:

In response to the Notification of Non-Compliant Appeal Brief (37 C.F.R. §41.37) dated December 13, 2004, enclosed is an amended Appeal Brief. The Notification states that the Brief does not contain a concise explanation of the subject matter defined in each of the independent claims involved in the appeal (see item 4 and item 10 of the Notification). However, the Brief filed on September 21, 2004, clearly had separate summaries for each of the independent claims. In

particular, page 3, line 18 through page 4, line 19, provide summaries for the independent claims under a heading entitled Summary of Claimed Subject Matter including references to the specification by page and line number and to the drawings by reference characters.

In a Telephone Interview on December 21, 2004 between Examiner Harkness and Appellants' representative, Examiner Harkness stated that the non-compliance would be corrected by an express identification of the respective independent claim number in each of the summaries. The amended Brief filed herewith expressly identifies by claim number each of the independent claims summarized in the Summary of Claimed Subject Matter. As such, the various concerns raised in the a Notice mailed December 13, 2004 have been addressed and the Brief should be entered.

If any additional fees are due, please charge Deposit Account No. 12-2252.

Respectfully submitted,

CHRISTOPHER P. MAIORANA, P.C.

Christopher P. Maiorana
Registration No. 42,829
24840 Harper Avenue, Suite 100
St. Clair Shores, MI 48080
(586) 498-0670

Dated: January 6, 2005

Docket No.: 00-162 / 1496.00047

Our Docket No.: 1496.00047

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re Application of:

Applicant	Ariel Cohen et al.		
Application No.:	09/746,796	Examiner:	Harkness, C.
Filed:	December 22, 2000	Art Group:	2183
For:	MICROCODE BASED HARDWARE TRANSLATOR TO SUPPORT MULTITUDE OF PROCESSORS		

I hereby certify that this letter, the response or amendment attached hereto are being deposited with the United States Postal Service as first class mail in an envelope addressed to Mail Stop Appeal Brief Patent, Commissioner for Patents, P.O. Box 1450, Alexandria, VA 22313-1450, on January 6, 2005.

By: Jan M Dunbar
Jan M. Dunbar

APPEAL BRIEF

Mail Stop Appeal Brief - Patents
Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

Dear Sir:

Appellant submits the following amended Appeal Brief pursuant to 37 C.F.R. §41.37(d) for consideration by the Board of Patent Appeals and Interferences. The amended Appeal Brief is submitted in response to a Notification of Non-Compliant Appeal Brief (37 CFR 41.37), mailed December 13, 2004. The fee for filing the opening brief was previously paid when the original Appeal Brief was first submitted. Therefore, no fee is due.

TABLE OF CONTENTS

I. REAL PARTY IN INTEREST

The real party in interest is the Assignee, LSI Logic Corporation.

II. RELATED APPEALS AND INTERFERENCES

There are no related appeals or interferences known to the Appellant, Appellant's legal representative, or Assignee which may be related to, directly affect or be directly affected by or have a bearing on the Board's decision in the pending appeal.

III. STATUS OF CLAIMS

Claims 1-22 are pending and remain rejected. The Appellant hereby appeals the rejection of claims 1-22 . A summary of the claims is attached in the Claim Appendix.

IV. STATUS OF AMENDMENTS

Appellant is appealing a final Office Action issued by the Examiner on May 18, 2004. No amendment has been made subsequent to the final rejection.

V. SUMMARY OF CLAIMED SUBJECT MATTER

In a first embodiment represented by independent claim 1, the presently claimed invention provides an apparatus comprising a circuit (100 in FIG. 3) configured to translate instruction codes of a first instruction set (page 11, line 15 - page 12, line 10) on-the-fly (page 27, lines 12-19) into addresses into a microcode memory (210 in FIG. 3 and page 27, lines 3-11)

containing sequences of instruction codes of a second instruction set that emulate a functionality of the instruction codes of the first instruction set (page 12, lines 2-10).

In a second embodiment represented by independent claim 17, the presently claimed invention provides an apparatus comprising (i) means for translating instruction codes of a first instruction set on-the-fly into addresses (208 in FIG. 3 and page 27, lines 3-11) into a microcode memory (210 in FIG. 3) containing sequences of instruction codes of a second instruction set that emulate a functionality of the instruction codes of said first instruction set (page 12, lines 2-10); (ii) means for receiving the instruction codes of the first instruction set (206 in FIG. 3) and (iii) means for presenting sequences of instruction codes of the second instruction set (126, 212 and 214 in FIG. 3, page 13, lines 6-8 and page 24, lines 5-16).

In a third embodiment represented by independent claim 18, the presently claimed invention provides a method for on-the-fly translation of instructions of a first instruction set into instructions of a second instruction set comprising the steps of: (A) receiving an instruction code of the first instruction set (input to 206 in FIG. 3 and page 20, lines 3-6); (B) generating an address into a microcode memory (210 in FIG. 3 and page 27, lines 3-11) in response to the instruction code of the first instruction set using a hardware translator (122 in FIG. 3), wherein the address points to a sequence of instruction codes of the second instruction set that will emulate the instruction code of the first instruction set (page 12, lines 6-8) and (C) presenting the sequence of instruction codes of the second instruction set (126, 212 and 214 in FIG. 3, page 13, lines 6-8 and page 24, lines 5-16).

VI. GROUNDS OF REJECTION TO BE REVIEWED ON APPEAL

Claims 1-6, 8, 9, 12-14 and 17-19 are rejected under 35 U.S.C. §102(b) as being anticipated by Hilgendorf et al.[1]

Claims 10, 11 and 20 are rejected under 35 U.S.C. §103(a) as being unpatentable over Hilgendorf in view of Martin.[2]

Claims 7 and 15 are rejected under 35 U.S.C. §103(a) as being unpatentable over Hilgendorf.

Claims 16 and 21 are rejected under 35 U.S.C. §103(a) as being unpatentable over Hilgendorf in view of Gee et al.[3]

---

[1] U.S. Patent No. 5,925,124 (hereinafter Hilgendorf).

[2] U.S. Patent No. 4,439,828.

[3] U.S. Patent No. 6,374,286 (hereinafter Gee).

VII. ARGUMENTS

A. Rejections under 35 U.S.C. § 102.

As set forth on page 2 of the final Office Action,[4] claims 1-6, 8, 9, 12-14 and 17-19 are rejected under 35 U.S.C. § 102(b) as being anticipated by Hilgendorf et al.[5]

The Federal Circuit has stated that "[t]o anticipate, *every element and limitation* of the claimed invention must be found in a single prior art reference, *arranged as in the claim.*"[6] The Federal circuit has added that the anticipation determination is viewed from one of ordinary skill in the art: "There must be no difference between the claimed invention and the reference disclosure, as viewed by a person of ordinary skill in the field of the invention."[7] Furthermore, "A claim is anticipated only if each and every element as set forth in the claim is found, either expressly or inherently described, in a single prior art reference."[8] As explained herein below, because Hilgendorf does not disclose or suggest a microcode memory containing sequences of instruction codes of a second instruction set that emulate a functionality of the instruction codes of the first instruction set, Hilgendorf does not anticipate the presently claimed invention.

---

[4] Dated May 18, 2004.

[5] U.S. Patent No. 5,925,124 (hereinafter Hilgendorf).

[6] *Brown v. 3M*, 60 USPQ2d 1375, 1376 (Fed. Cir. 2001) citing *Karsten Mfg. Corp. v. Cleveland Golf Co.*, 242 F.3d 1376, 1383, 58 USPQ2d 1286, 1291 (Fed. Cir. 2001); *Scripps Clinic & Research Found. v. Genentech Inc.*, 927 F.2d 1565, 18 U.S.P.Q.2d 1001, 1010 (Fed. Cir. 1991) (Emphasis added by Appellant).

[7] *Scripps Clinic & Research Found. v. Genentech Inc.*, 927 F.2d 1565, 18 U.S.P.Q.2d 1001, 1010 (Fed. Cir. 1991).

[8] *Verdegaal Bros. V. Union Oil Co. of California*, 814 F.2d 628, USPQ2d 1051, 1053 (Fed Circ. 1987).

1. Claims 1-3, 5-9, 12-14 and 17-19 are fully patentable over Hilgendorf.

The presently pending claim 1 provides an apparatus comprising a circuit configured to translate instruction codes of a first instruction set on-the-fly into addresses into a microcode memory containing sequences of instruction codes of a second instruction set that emulate a functionality of the instruction codes of the first instruction set. Claims 17 and 18 include similar limitations. Claims 2, 3, 5-9, 12-14 and 19 depend, directly or indirectly, from either claim 1 or claim 19.

Hilgendorf does not disclose or suggest a microcode memory containing sequences of instruction codes of a second instruction set that emulate a functionality of the instruction codes of the first instruction set, as presently claimed. Therefore, Hilgendorf does not disclose or suggest each and every element of the presently claimed invention, arranged as in the claims. As such, claims 1-3, 5-9, 12-14 and 17-19 are fully patentable over the cited reference and the rejection should be reversed.

Hilgendorf is directed to dynamic conversion between different instruction codes by **recombination** of instruction elements.[9] In particular, Hilgendorf refers to **converting** microcode instructions of a code A into internal instructions of a code B, **not translating** instruction codes of a first instruction set on-the-fly **into addresses into a microcode memory containing sequences of instruction codes of a second instruction set** that emulate a functionality of the instruction codes of the first instruction set, as presently claimed. Specifically, Hilgendorf states:

> There exist two possible methods for applying the instruction conversion mechanism described **to microcode instructions**. The

[9] Title of Hilgendorf, emphasis added.

> first method is to relate each microcode instructions of code A to a huge number of explicitly defined internal instructions of code B. Of course, these internal instructions would have to be generated sequentially.
>
> The second method is to first **translate said microcode instructions of code A** to the corresponding sequence of hard-wired instructions of code A. Then, in a next step, each of the hard-wired external instructions has to be translated to the corresponding sequence of internal instructions by means of the translation table 106. The advantage of this second solution is that for each hard-wired external instruction, the corresponding internal instructions can be generated in parallel (column 11, lines 53-67 of Hilgendorf, emphasis added).

Assuming, *arguendo*, the translation table 106 of Hilgendorf is similar to the presently claimed microcode memory,[10] Hilgendorf does not disclose or suggest each and every element of the presently claimed invention, arranged as in the claims. Specifically, the cited reference clearly explains that the translation table 106 does not contain sequences of instruction codes of a second instruction set that emulate a functionality of the instruction codes of the first instruction set, as presently claimed. Rather, Hilgendorf clearly states that the translation table contains rearrangement information.[11] Furthermore, Hilgendorf is silent about **a microcode memory containing sequences of instruction codes of a second instruction set that emulate a functionality of the instruction codes of the first instruction set**, as presently claimed. Therefore, Hilgendorf does not disclose or suggest each and every element of the presently claimed invention, arranged as in the present claims. As such, the presently claimed invention is fully patentable over the cited reference and the rejection should be withdrawn.

---

[10] As suggested by the Examiner in section 5 on page 2 of the final Office Action dated May 18, 2004 and for which Appellant's representative does not necessarily agree.

[11] See Abstract and column 7, lines 9-30 of Hilgendorf.

Furthermore, rather than **translating** instruction codes of a first instruction set on-the-fly **into addresses into a microcode memory** containing sequences of instruction codes of a second instruction set as presently claimed, Hilgendorf **converts** instructions of a code A (i.e., external instructions 100 of Hilgendorf) into instructions of a code B (i.e., internal instructions 101 or 400 of Hilgendorf) **by rearranging elements** of the instructions of the code A **using multiplexers** (see, for example, FIGS. 1 and 5 and abstract of Hilgendorf). Since Hilgendorf (i) is silent regarding a microcode memory containing sequences of instruction codes of a second instruction set that emulate a functionality of the instruction codes of the first instruction set, as presently claimed, and (ii) **converts** instructions of a code A into instructions of a code B **by rearranging elements** of the instructions of the code A using multiplexers, Hilgendorf does not disclose or suggest a circuit configured to **translate instruction** codes of a first instruction set on-the-fly **into addresses into a microcode memory** containing sequences of instruction codes of a second instruction set that emulate a functionality of the instruction codes of the first instruction set, as presently claimed. Therefore, Hilgendorf does not disclose or suggest each and every element of the presently claimed invention, arranged as in the present claims. As such, the claims 1-3, 5-9, 12-14 and 17-19 are fully patentable over the cited reference and the rejection should be reversed.

Furthermore, the Examiner's position that (i) "the opcode of the first instruction set is translated to addresses used to indicate a set of internal instruction in the translation table, which is implemented using memory, thus the translation table is a microcode memory"[12] and (ii) the translation table 106 contains sequences of instruction codes of a second instruction set that emulates

---

[12] See section 5, lines 4-6, on page 2 of the final Office Action dated May 18, 2004.

a functionality of the instruction codes of the first instruction set[13] is not support by the cited reference. Specifically, Hilgendorf is directed to an apparatus and a method for converting instructions of a code A to instructions of a code B.[14] The conversion is performed "by obtaining rearrangement information which corresponds to the instruction that is to be converted from a table."[15] The rearrangement information is then used to rearrange the instruction elements of the initial instruction in order to generate instructions of code B which functionally corresponds to the initial instruction.[16] The cited reference clearly explains that the translation table 106 does not contain sequences of instruction codes of a second instruction set that emulate a functionality of the instruction codes of the first instruction set, as presently claimed. Rather, Hilgendorf clearly states that the translation table contains rearrangement information.[17]

Furthermore, Hilgendorf further teaches that the rearrangement information is used to control multiplexing means which use the instruction elements of the initial code A instruction as input.[18] A person of ordinary skill in the field of the invention would not consider a translation table containing rearrangement information for controlling multiplexers, as taught by Hilgendorf, to be the same as a microcode memory containing sequences of instruction codes of a second

---

[13] As suggested by the Examiner in section 5, lines 1-10, on page 2 of the final Office Action dated May 18, 2004.

[14] Abstract of Hilgendorf.

[15] Abstract of Hilgendorf.

[16] Abstract of Hilgendorf.

[17] See Abstract and column 7, lines 9-30 of Hilgendorf.

[18] See Abstract and column 7, lines 35-45 of Hilgendorf.

instruction set that emulate a functionality of the instruction codes of the first instruction set, as presently claimed. Furthermore, the Examiner failed to make the required showing by presenting any evidence or a convincing line of reasoning why a person of ordinary skill in the field of the invention would view Hilgendorf as having no differences from the presently claimed invention. [19] Therefore, the Examiner has failed to meet the Office's burden to factually establish a *prime facie* case of anticipation. As such, claims 1-3, 5-9, 12-14 and 17-19 are fully patentable over the cited reference and the rejection should be reversed.

2. Claim 4 is fully patentable over Hilgendorf.

The presently pending claim 4 depends directly from claim 1 and, therefore, includes all the limitations of claim 1. Consequently, the arguments presented above in support of claim 1 are incorporated herein by reference in support of claim 4. Claim 4 further recites that predetermined sequences of the instruction codes of the first instruction set are used to address the microcode memory. In contrast, the portions of Hilgendorf cited by the Examiner do not provide evidence or a convincing line of reasoning that Hilgendorf discloses or suggests that predetermined sequences of the instruction codes of the first instruction set are used to address the microcode memory, as presently claimed.

Specifically, the Examiner cites the Abstract, FIGS. 1 and 2, column 3, lines 8-56 and column 7, lines 9-30 as support for the conclusory statement that "Hilgendorf has taught wherein predetermined sequences of said instruction codes of the first instruction set are used to address the

[19] See section 5, lines 1-10 on page 2 of the final Office Action dated May 18, 2004.

microcode memory."[20] In particular, the Abstract of Hilgendorf cited by the Examiner reads:

> The invention provides an apparatus and a method for converting instructions of a code A to instructions of a code B. Said conversion is performed by obtaining rearrangement information, which corresponds to **the instruction that is to be converted**, from a table. Said rearrangement information is then used to rearrange the instruction elements of **the initial instruction**, in order to generate instructions of code B, which functionally corresponds to **said initial instruction**. Said rearrangement can be performed by multiplexing means, which use said instruction elements of **the initial code A instruction** as input, and which select one of said instruction elements, or the content of another register, and forward this selected data to the instruction that is to be generated. Said rearrangement information is directly used to control the selection performed by said multiplexers (Abstract of Hilgendorf, emphasis added).

The Abstract of Hilgendorf is silent regarding a sequence of the instruction codes of the first instruction set being used to address the microcode memory, as presently claimed. Rather, each reference to an instruction of code A is singular, thus indicating only one instruction of code A is involved at a time.

Another portion of Hilgendorf cited by the Examiner reads:

> The invention provides both an apparatus and a method for converting instructions of a code A to instructions of a code B, whereby each of said instructions of code A and each of said instructions of code B consist of instruction elements.
>
> According to the invention, rearrangement information is obtained, in a first step, from at least one table, said rearrangement information corresponding to **an instruction** of said instructions of code A that is to be converted to code B. Said instruction elements of said instruction of code A are rearranged according to said rearrangement information, in a second step, in order to generate at least one instruction of said instructions of code B.
>
> Said rearrangement is done by rearrangement means, which use instruction elements of **said instruction of code A** as a first input

[20] See section 8 on page 3 of the final Office Action dated May 18, 2004.

and use said rearrangement information corresponding to **said instruction of code A** as a second input, and which rearrange said instruction elements of said instruction of code A according to said rearrangement information in order to generate at least one instruction of said instructions of code B functionally corresponding to said instruction of code A.

The initial **instruction of code A** already contains the identifiers of logical registers that are necessary for generating an instruction of code B. Instead of rearranging the OP-codes, the register identifiers, status information, addresses etc. by logic means in order to generate an instruction of code B, the recombination is performed according to a pattern in a table that is related to said initial **instruction of code A**.

This provides for a fast instruction translation, which only requires one cycle, because the bit pattern stored in said table can directly be used by a hardware to perform the correct routing of logical register addresses, status bits, logical register identifiers, etc. This only takes one cycle, and therefore, instruction translation and decoding can be performed at run time.

Another advantage of the invention is that the rearrangement information contained in said table means can easily be exchanged. In case there exist any errors or inconsistencies of the way instruction elements of the initial instruction are routed to the new instruction, these errors can easily be fixed by changing the table accordingly. In case of an update of either the architected code A or code B, or in case new features and/or instructions are included in any of both codes, the necessary changes can easily be made to said table means. An important point is that said changes can be implemented selectively, which means that only the instructions that have been changed have to be updated (column 3, lines 8-56 of Hilgendorf, emphasis added).

Again, Hilgendorf refers to using a single instruction of code A as an input to the conversion process.

Yet another portion of Hilgendorf cited by the Examiner reads:

Rearrangement information needed for converting **the external instruction** (100) to a set of internal instructions (101) is contained in a translation table (106). Said translation table can either be implemented as a ROM table or as a RAM table. **Each entry (105) of said table corresponds to one of said external code A instructions** (100), and determines how the instruction elements of

> this specific instruction are to be rearranged in order to form the corresponding internal instructions (101).
>
> As **each translation table entry (105) corresponds to one external code A instruction (100), the OP-code (102) of said external instruction can be used to determine the correct translation table entry**. This is done by forwarding the OP-code (102) to an address generation logic (104), which converts said OP-code to the address of the corresponding entry in the table. This address is then used to access (103) the corresponding translation table entry (105). Said entry contains the OP-codes of all the internal code B instructions (107) to which the external instruction is to be converted to, and multiplexer control information (129), which is used for controlling the recombination of the external instruction's elements (column 7, lines 9-30 Hilgendorf, emphasis added).

Since each translation table entry corresponds to ONE external code A instruction, it follows that, based on the Examiner's own citation, Hilgendorf does not disclose or suggest that predetermined sequences of the instruction codes of the first instruction set are used to address the microcode memory, as presently claimed. Thus, Hilgendorf does not disclose or suggest each and every element of the presently claimed invention, arranged as in the present claims. Therefore, the Examiner has failed to meet the Office's burden to factually establish a *prime facie* case of anticipation. As such, claim 4 is fully patentable over the cited reference and the rejection should be reversed.

B. <u>Rejections under 35 U.S.C. § 103</u>

As set forth on pages 7-9 of the final Office Action,[21] claims 7, 10, 11, 15, 16, 20 and 21 are rejected under 35 U.S.C. §103(a) as being unpatentable over either Hilgendorf alone (claims

[21] Dated May 18, 2004.

7 and 15), Hilgendorf in combination with Martin (claims 10, 11 and 20) or Hilgendorf in combination with Gee (claim 16 and claim 21).

The Examiner bears the initial burden of factually supporting any *prima facie* conclusion of obviousness.[22] If the Examiner does not produce a *prima facie* case, the Applicant is under no obligation to submit evidence of nonobviousness.[23] "[T]o establish obviousness based on a combination of the elements disclosed in the prior art, there must be some motivation, suggestion or teaching of the desirability of making the specific combination that was made by the applicants."[24] "[T]he factual inquiry whether to combine references must be thorough and searching."[25] "This factual question ... [cannot] be resolved on subjective belief and unknown authority."[26] "It must be based on objective evidence of record."[27] The Examiner must show that (a) there is some suggestion or motivation, either in the references or in the knowledge generally available to one of ordinary skill in the art, to modify or combine the references, (b) there is a reasonable expectation of success, and

---

[22] Manual of Patent Examining Procedure (M.P.E.P.), Eighth Edition, Rev. 2, May 2004, §2142.

[23] *Id.*

[24] *In re Kotzab*, 217 F.3d 1365, 1370, 55 USPQ2d 1313, 1316 (Fed. Cir. 2000) (citing *In re Dance*, 160 F.3d 1339, 1343, 48 USPQ2d 1635, 1637 (Fed. Cir. 1998); *In re Gordon*, 733 F.2d 900, 902, 221 USPQ 1125, 1127 (Fed. Cir. 1984)).

[25] *McGinley v. Franklin Sports, Inc.*, 262 F.3d 1339, 1351-52, 60 USPQ2d 1001, 1008 (Fed. Cir. 2001).

[26] *In re Lee*, 277 F.3d 1338, 1343-44, 61 USPQ2d 1430, 1434 (Fed. Cir. 2002).

[27] *Id.* at 1343, 61 USPQ2d at 1434.

(c) the prior art reference (or combination of references) teaches or suggests all of the claim limitations.[28]

The Federal Circuit has held that both the suggestion to modify or combine the references and the reasonable expectation of success must be found in the prior art itself, not merely in Appellant's disclosure.[29] Furthermore, the Court of Appeals for the Federal Circuit has indicated that the requirement for showing the teaching of motivation to combine references is "rigorous" and must be "clear and particular."[30] Furthermore, the Board has held that the claimed invention is obvious only if either the references expressly or implicitly suggest the claimed invention, or **a convincing line of reasoning is presented by the examiner as to why an artisan would have found the claimed invention to be obvious in light of the teachings of the cited references.**[31]

As explained herein below, the Examiner has failed to establish a *prima facie* case of obviousness (i) because the cited references, alone or in combination, fail to teach or suggest all the elements of the presently claimed invention, (ii) because the Examiner has failed to provide a clear and particular suggestion or motivation, either in the references or in the knowledge generally available to one of ordinary skill in the art, to modify and/or combine the references and (iii) because the Examiner failed to put forth a convincing line of reasoning as to why an artisan would have

---

[28] M.P.E.P. §2142.

[29] See *In re Vaeck*, 947 F.2d 488, 20 U.S.P.Q.2d 1438, 1442 (Fed. Cir. 1991).

[30] *In re Anita Dembiczak* and *Benson Zinbarg*, 50 U.S.P.Q.2d 1614 (Fed. Cir. 1999).

[31] See *Ex Parte Clapp*, 227 U.S.P.Q. 972, 973 (Bd. Pat. App. & Inter. 1985) (emphasis added by Appellant).

found the claimed invention to be obvious in light of the teachings of the cited references.[32] As such, the presently pending claims 7, 10, 11, 15, 16, 20 and 21 are fully patentable over the cited references and the rejection should be reversed.

1. Claims 10, 11 and 20 are fully patentable over Hilgendorf and Martin.

Claim 10 depends directly from claim 1 and, therefore, includes all of the limitations of claim 1. Consequently, the arguments presented above in support of claim 1 are incorporated herein by reference in support of claim 10. Claim 10 further provides that the circuit is configured to detect optimizable sequences of instruction codes on-the-fly. Claim 20 includes a similar limitation. Claim 11 depends directly from claim 10.

The Examiner admits that with respect to claims 10, 11 and 20 Hilgendorf has not taught wherein the circuit is configured to detect optimizable sequences of instruction codes on-the-fly. Martin does not appear to cure the deficiencies of Hilgendorf. In particular, Martin appears to be silent regarding a circuit configured to detect optimizable sequences of instruction codes on-the-fly. Specifically, the text of Martin cited by the Examiner as support for the position that Martin "has taught a circuit configured to detect optimizable sequences of instruction codes on-the-fly"[33] reads:

> Buffered, pre-fetched instructions in the instruction handling portion of a data processing system are examined to detect sequences of predetermined instructions to effect generation of a substitute instruction to be executed by an execution unit in place of the first

---

[32] M.P.E.P. §2142.

[33] See section 21, lines 2-4 on page 7 of the final Office Action dated May 18, 2004.

instruction of the sequence to perform the functions specified by all of the instructions of the sequence.[34]

> The present invention improves the performance of a data processing system with instruction decode and execute overlap without change to the architecture, the instruction set, the operating systems, the compilers, or the vast inventory of running production programs. Certain pre-defined instruction sequences are detected, and a substitute instruction is generated to replace the first instruction of the sequence, causing the execution unit of the data processing system to respond to the single substitute instruction to perform all the functions called for by the original sequence of instructions.
>
> An instruction counter accesses instructions from an instruction buffer for transfer to an instruction register to be decoded and address generation performed. A sequence detector examines the Op-code field of adjacent instructions in the instruction buffer, and various 4-bit fields of adjacent instructions, to determine the presence of particular predefined instruction sequences. Detection of a particular sequence generates a signal to a substitute instruction generator which is effective to modify the first instruction of the instruction sequence in the instruction buffer. The substitute instruction will be identified by a unique Op-code field which defines a "pseudo instruction", in that it is not found in the assigned Op-codes of the instruction set. In a data processing system which has an execution unit controlled by microprogramming in a control store, the execution of the pseudo instruction can be effected by storing suitable microcode in the control store for access by the Op-code of the pseudo instruction to control all of the functions required of the execution unit to realize the results of all of the functions called for by the original sequence.[35]

Nowhere in the above text does Martin expressly mention detecting optimizable sequences of instruction codes **on-the-fly**, as presently claimed.

The Examiner has provided a personal conclusion that "It would have been obvious to one of ordinary skill in the art at the time of the invention to be configured to detect optimizable

---

[34] Abstract of Martin, emphasis added.

[35] Column 2, lines 20-52 of Martin, emphasis added.

sequences of instruction codes on-the-fly. Since Martin shows us that one instruction can take the place of two or more instructions, an that this function improves performance"[36] Such a conclusory statement is not adequate to meet the burden to identify specifically the reasons one of ordinary skill in the art would have been motivated to select the references and combine them.[37]

Coupled with this provision is the additional requirement that the suggestion or motivation exist before the date of invention.[38] Thus, it is incorrect for the Examiner to formulate the suggestion or motivation based on current knowledge; the Examiner must remove all knowledge that he or she has accumulated since the date of invention.[39] As stated by the Federal Circuit:

> It is impermissible to use the claimed invention as an instruction manual or "template" to piece together the teachings of the prior art so that the claimed invention is rendered obvious. This court has previously stated that "[o]ne cannot use hindsight reconstruction to pick and choose among isolated disclosures in the prior art to deprecate the claimed invention."[40]

---

[36] Lines 4-7 of section 21 on page 7 of the final Office Action dated May 18, 2004.

[37] *In re Lee*, 277 F.3d 1338, 1343-44, 61 USPQ2d 1430, 1434 (Fed. Cir. 2002) ("The examiner's conclusory statements . . . do not adequately address the issue of motivation to combine.); In re Rouffet, 149 F.3d 1350, 1359, 47 U.S.P.Q.2D (BNA) 1453, 1459 (Fed. Cir. 1998) ("the Board must identify specifically . . . the reasons one of ordinary skill in the art would have been motivated to select the references and combine them").

[38] 35 U.S.C. §103(a) (Supp. 1995) ("differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole *would have been* obvious at the time the invention was made"). Emphasis added by Appellant.

[39] *Panduit Corp. v. Dennison Mfg. Co.*, 810 F.2d 1561, 1 USPQ 2d 1593, 1595-96 (Fed.Cir.) ("To reach a proper conclusion under Section 103, the decisionmaker must step backward in time and into the shoes worn by that 'person' when the invention was unknown and just before it was made."), *cert. denied*, 481 U.S. 1052 (1987).

[40] *In re Fritch*, 972 F.2d 1260, 23 USPQ 2d 1780, 1784 (Fed.Cir. 1992) (quoting *In re Fine*, 837 F.2d 1071, 1075, 5 USPQ 2d 1596, 1600 (Fed.Cir. 1988)).

The Federal Circuit has made it clear that the best defense against the subtle but powerful attraction of a hindsight-based obviousness analysis is rigorous application of the requirement for a showing of the teaching or motivation to combine prior art references.[41] Failure of the Examiner to provide the necessary suggestion or motivation creates a presumption that the combination of references selected by the Examiner to support the obviousness rejection were based on hindsight.[42] The invention does not make itself obvious; that suggestion or teaching must come from the prior art.[43]

The Examiner's conclusory statement that "It would have been obvious to one of ordinary skill in the art at the time of the invention to be configured to detect optimizable sequences of instruction codes on-the-fly. Since Martin shows us that one instruction can take the place of two

---

[41] See, e.g., *C.R. Bard, Inc. v. M3 Sys., Inc.*, 157 F.3d 1340, 1352, 48 USPQ 2d 1225, 1232 (Fed.Cir. 1998) (describing "teaching or suggestion or motivation [to combine]" as an "essential evidentiary component of an obviousness holding"); *In re Rouffet*, 149 F.3d 1350, 1359, 47 USPQ 2d 1453, 1459 (Fed.Cir. 1998) ("the Board must identify specifically ... the reasons one of ordinary skill in the art would have been motivated to select the references and combine them").

[42] *In re Rouffet*, 149 F.3d 1350, 47 USPQ 2d 1453, 1458 (Fed.Cir. 1998) (Because the Board did not explain the specific understanding or principle within the knowledge of a skilled artisan that would motivate one with no knowledge of Rouffet's invention to make the combination, this court infers that the examiner selected these references with the assistance of hindsight. This court forbids the use of hindsight in the selection of references that comprise the case of obviousness. See *In re Gorman*, 933 F.2d 982, 986, 18 U.S.P.Q.2D (BNA) 1885, 1988 (Fed.Cir. 1991). Lacking a motivation to combine references, the Board did not show a proper prima facie case of obviousness. This court reverses the rejection over the combination of King, Rosen, and Ruddy.).

[43] See, e.g., *Uniroyal, Inc. v. Rudkin-Wiley Corp.*, 837 F.2d 1044, 1051-52, 5 USPQ 2d 1434, 1438 (Fed.Cir. 1988) (it is impermissible to reconstruct the claimed invention from selected pieces of prior art absent some suggestion, teaching, or motivation in the prior art to do so); *Interconnect Planning Corp. v. Feil*, 774 F.2d 1132, 1143, 227 USPQ 543, 551 (Fed.Cir. 1985) (it is insufficient to select from the prior art the separate components of the inventor's combination, using the blueprint supplied by the inventor); *Fromson v. Advance Offset Plate, Inc.*, 755 F.2d 1549, 1556, 225 USPQ 26, 31 (Fed.Cir. 1985) (the prior art must suggest to one of ordinary skill in the art the desirability of the claimed combination).

or more instructions, an that this function improves performance"[44] does not adequately address the issue of motivation.[45] Specifically, Martin states that "The present invention improves the performance of **a data processing system with instruction decode and execute overlap**."[46] However, the Examiner has made no findings and presented no evidence or convincing line of reasoning that one of ordinary skill in the art would recognize Hilgendorf as applicable to **a data processing system with instruction decode and execute overlap**. It is improper, in determining whether a person of ordinary skill would have been led to a combination of references, simply to use that which the inventor taught against its teacher.[47] Furthermore, the Examiner failed to present any evidence or convincing line or reasoning with respect to the issue of a reasonable expectation of success in combining Hilgendorf and Martin.[48] Therefore, because the Examiner failed to make the required showings with regard to motivation and the reasonable expectation of success, the Examiner failed to meet the Office's burden to factually establish a *prima facie* conclusion of obviousness.[49] As such, claims 10, 11 and 20 are fully patentable over the cited references and the rejection should be reversed.

---

[44] Lines 4-7 of section 21 on page 7 of the final Office Action dated May 18, 2004.

[45] *In re Lee*, 277 F.3d 1338, 1343-44, 61 USPQ2d 1430, 1434 (Fed. Cir. 2002)

[46] Column 2, lines 19-21 of Martin.

[47] *In re Lee*, 277 F.3d 1338, 1343-44, 61 USPQ2d 1430, 1434 (Fed. Cir. 2002) citing *W.L. Gore v. Garlock, Inc.*

[48] See section 21 on page 7 of the final Office Action dated My 18, 2004.

[49] M.P.E.P. §2142.

2. Claim 15 is fully patentable over Hilgendorf

Claim 15 depends directly from claim 1 and, therefore, includes all of the limitations of claim 1. Consequently, the arguments presented above in support of claim 1 are incorporated herein by reference in support of claim 15. Claim 15 further provides that the instruction codes of the first instruction set comprise Java bytecodes.

The Examiner admits that with respect to claim 15 Hilgendorf has not explicitly taught wherein said instruction codes of said first instruction set comprise Java bytecodes.[50] The Examiner's conclusory statement that:

> Since Java is a commonly used programing language, one of ordinary skill in the art at the time of the invention would have recognized that you could execute Java code on a computer using Hilgendorf's invention by translating the code to the host's instruction code. Therefore, it would have been obvious to one of ordinary skill in the art at the time of the invention to translate Java code into the host's computer code since Java is a popular and widely used programming language and many programs would already be written in Java.[51]

does not adequately address the issue of motivation. Specifically, the Examiner failed to present evidence or provide a convincing line of reasoning explaining exactly how Hilgendorf's invention could be used with Java code. In particular, Hilgendorf's invention is directed to dynamic conversion between different instruction codes by recombination of instruction elements. However, Hilgendorf appears to silent regarding converting and executing Java code. Specifically, the text of Hilgendorf cited by the Examiner reads:

---

[50] See section 24 on page 8 of the final Office Action dated May 18, 2004.

[51] Page 8, section no. 24 of the Final Office Action dated May 18, 2004.

> Another advantage of the invention is that the rearrangement information contained in said table means can easily be exchanged. In case there exist any errors or inconsistencies of the way instruction elements of the initial instruction are routed to the new instruction, these errors can easily be fixed by changing the table accordingly. In case of an update of either the architected code A or code B, or in case new features and/or instructions are included in any of both codes, the necessary changes can easily be made to said table means. An important point is that said changes can be implemented selectively, which means that only the instructions that have been changed have to be updated.[52]
>
> The possibility of converting one set of RISC instructions to a different set of RISC instructions allows to process an instruction stream of code A by a processor which normally can only handle instruction code B. By means of translation tables containing rearrangement information, one processor can process a variety of different RISC codes which are not the architected codes of said processor. By either using different tables in parallel or by exchanging the contents of a table, it is possible to switch between different codes. As code conversion is achieved at run time, one processor can "understand" a lot of different RISC code flavors.[53]

Nowhere in the above text, or elsewhere, does Hilgendorf expressly mention converting or executing Java code.

Furthermore, Hilgendorf is directed to rearranging elements of one instruction code to generate another instruction code. The Examiner has failed to present any evidence or convincing line of reasoning to factually support a conclusion that Java bytecodes can merely be rearranged, as performed by Hilgendorf, to generate host instructions. Furthermore, the Examiner failed to present any evidence or convincing line of reasoning with regard to the reasonable expectation of success in converting Java bytecodes to the host's instruction code using the teachings of Hilgendorf. The

---

[52] Column 3, lines 45-56 of Hilgendorf.

[53] Column 4, lines 51-61 of Hilgendorf).

conclusory statement that it would have been obvious to one of ordinary skill in the art at the time of the invention to translate Java code into the host's computer code since Java is a popular and widely used programming language, and many programs would already be written in Java"[54] does not adequately address the issue of the reasonable expectation of success.

Thus, because the Examiner failed to make the required showing that there is a reasonable expectation of success in using the invention of Hilgendorf with Java bytecodes, the Examiner failed to factually establish a *prima facie* conclusion of obviousness.[55] As such, the presently pending claim 15 is fully patentable over the cited references and the rejection should be reversed.

3. Claim 16 is fully patentable over Hilgendorf and Gee

Claim 16 depends directly from claim 1 and, therefore, includes all of the limitations of claim 1. Consequently, the arguments presented above in support of claim 1 are incorporated herein by reference in support of claim 16. Claim 16 further provides that the circuit comprises a portion of a Java virtual machine implemented in hardware.

The Examiner admits that Hilgendorf has not explicitly taught wherein said circuit comprises a Java virtual machine implemented in hardware.[56] The Examiner takes the position that

---

[54] Lines 7-10 in section 24 on page 8 of the final Office Action dated May 18, 2004.

[55] M.P.E.P. §2142.

[56] Page 8, section 26, lines 1-2 of the final Office Action dated May 18, 2004.

Gee has taught wherein said circuit comprises a portion of a Java virtual machine implemented in hardware.[57] The text of Gee cited by the Examiner in support of this position reads:

> A program written in the JAVA language is compiled from the JAVA source code into as a series of platform independent commands called "bytecodes." Generally, a JAVA host computer system utilizes a JAVA run time environment to interpret and execute the bytecodes. The JAVA run time environment is called a "JAVA virtual machine" (JVM) and it can either interpret the bytecodes directly or use a "just-in-time" compiler to convert the bytecodes to platform dependent codes which run directly on the host platform. The JVM allows a flexible run-time environment.
>
> However, in real-time embedded processor applications, the JVM has some significant drawbacks including speed and difficulty in meeting the strict timing requirements normally found in such environments. Consequently, in such applications, another type of processor is often used in which the JAVA bytecodes generated by a JAVA compiler are executed directly on the platform. In this case, the JAVA bytecodes are the low level assembly language of the processor and the JAVA program could be said to be running on a "JAVA machine" rather than on a JVM. Such processors are called "direct execution" JAVA processors.
>
> In the direct execution JAVA processor context, the term JAVA Virtual Machine refers to a logical address space for JAVA classes within which one or more threads can execute. Within such a logical address space, several JAVA applications can concurrently run. Each JAVA application may also create multiple class name spaces by creating one or more class loaders, but, in the absence of application-created class loaders, only one class name space is created by the default class loader in a JVM (column 2, lines 38-67 of Gee).

Nowhere in the cited text does Gee expressly mention a circuit comprises **a portion of a Java virtual machine implemented in hardware**, as presently claimed.

Furthermore, the Examiner failed to explain the specific understanding or principle within the knowledge of a person of ordinary skill in the art, with no knowledge of the presently

---

[57] Page 8, section 26, lines 2-3 of the final Office Action dated May 18, 2004.

claimed invention, that would have led to the selection and combination of Hilgendorf and Gee.[58] Furthermore, the Examiner failed to present any evidence or convincing line of reasoning regard a reasonable expectation of success. Thus, the Examiner failed to meet the Office's burden of factually establishing a *prima facie* conclusion of obviousness.[59] As such, the presently pending claim 16 is fully patentable over the cited reference and the rejection should be reversed.

4. Claim 21 is fully patentable over Hilgendorf and Gee

Claim 21 depends directly from claim 1 and, therefore, includes all of the limitations of claim 1. Consequently, the arguments presented above in support of claim 1 are incorporated herein by reference in support of claim 21. Claim 21 further provides that the sequences of instruction codes of the second instruction set comprise one or more virtual stack references.

The Examiner admits that Hilgendorf has not taught wherein said sequence of instruction codes of said instruction set comprise one or more virtual stack references.[60] The Examiner takes the position that Gee teaches wherein said sequence of instruction codes of said

---

[58] *In re Rouffet*, 149 F.3d 1350, 47 USPQ 2d 1453, 1458 (Fed.Cir. 1998) (Because the Board did not explain the specific understanding or principle within the knowledge of a skilled artisan that would motivate one with no knowledge of Rouffet's invention to make the combination, this court infers that the examiner selected these references with the assistance of hindsight. This court forbids the use of hindsight in the selection of references that comprise the case of obviousness. See *In re Gorman*, 933 F.2d 982, 986, 18 U.S.P.Q.2D (BNA) 1885, 1988 (Fed.Cir. 1991). Lacking a motivation to combine references, the Board did not show a proper prima facie case of obviousness. This court reverses the rejection over the combination of King, Rosen, and Ruddy.).

[59] M.P.E.P. §2142.

[60] Page 9, section 27, lines 1-2 in the final Office Action dated May 18, 2004.

instruction set comprise one or more virtual stack references.[61] The text of Gee cited by the Examiner in support of this position reads:

> The JVM is actually composed of one or more threads. Each JVM thread has a private JAVA stack, created at the same time as the thread, which stores JVM frames. A JAVA stack is the equivalent of [the] stack of a conventional programming language such as C. The JAVA stack holds local variables and partial results, and plays a part in method invocation and return. The JVM specification permits JAVA stacks to be of either a fixed or a dynamically varying size. The JVM also has a method area that is shared among all threads (column 7, lines 1-10 of Gee).

Nowhere in the above text does Gee expressly mention a sequence of instruction codes of an instruction set comprises one or more virtual stack references, as presently claimed.

Furthermore, the Examiner failed to explain the specific knowledge or understanding in the art that would have compelled a person of ordinary skill in the art, with no knowledge of the presently claimed invention, to select and combine Hilgendorf and Gee.[62] Furthermore, the Examiner failed to present any evidence or convincing line of reasoning regarding a reasonable expectation of success.[63] Thus, the Examiner failed to meet the Office's burden of factually establishing a *prima facie* conclusion of obviousness.[64] As such, the presently pending claim 21 is fully patentable over the cited references and the rejection should be reversed.

---

[61] Page 9, section 27, lines 2-4 in the final Office Action dated May 18, 2004.

[62] See page 9, section 27, lines 1-12 in the final Office Action dated May 18, 2004.

[63] *Id.*

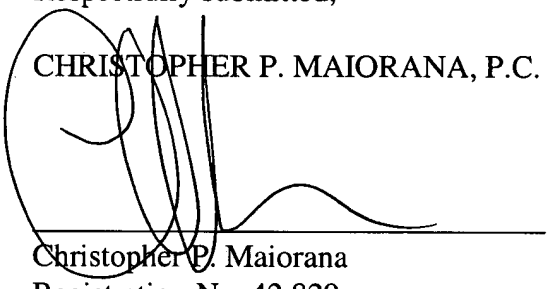
[64] M.P.E.P. §2142.

C. CONCLUSION

The cited references do not disclose or suggest a microcode memory containing sequences of instruction codes of a second instruction set that emulate a functionality of the instruction codes of the first instruction set, as recited in claims 1, 17 and 18. Hence, the Examiner has clearly erred with respect to the patentability of the claimed invention. It is respectfully requested that the Board overturn the Examiner's rejection of all pending claims, and hold that the claims are not rendered anticipated or obvious by the cited reference. However, should the Board find the arguments herein in support of independent claims 1, 17 and/or 18 unpersuasive, the Board is respectfully requested to carefully consider the arguments set forth above in support of dependent claims 4, 10, 11, 15, 16, 20 and 21.

Respectfully submitted,

CHRISTOPHER P. MAIORANA, P.C.



Christopher P. Maiorana
Registration No. 42,829

Dated: January 6, 2005

c/o
Intellectual Property Law Department
LSI Logic Corporation
1621 Barber Lane
MS: D-106 Legal
Milpitas, CA 95035

VIII. CLAIM APPENDIX

The claims of the present application which are involved in this appeal are as follows:

1. An apparatus comprising:

a circuit configured to translate instruction codes of a first instruction set on-the-fly into addresses into a microcode memory containing sequences of instruction codes of a second instruction set that emulate a functionality of the instruction codes of the first instruction set.

2. The apparatus according to claim 1, wherein said sequences of instruction codes of said second instruction set generated in response to said instruction codes of said first instruction set are stored in a cache.

3. The apparatus according to claim 1, wherein said circuit comprises a decoder configured to generate said addresses into said microcode memory.

4. The apparatus according to claim 1, wherein predetermined sequences of said instruction codes of said first instruction set are used to address said microcode memory.

5. The apparatus according to claim 1, wherein addresses into said microcode memory are generated by a look-up-table in response to said instruction codes of said first instruction set.

6. The apparatus according to claim 1, wherein said instruction codes of said second instruction set comprise native instructions of a target processor.

7. The apparatus according to claim 6, wherein said target processor is selected from the group consisting of MIPS, ARM, and Motorola 68K.

8. The apparatus according to claim 3, wherein said microcode memory can be reprogrammed to support different processors.

9. The apparatus according to claim 1, wherein said circuit is configured to format the sequences of instruction codes of said second instruction set according to an opcode format of a processor.

10. The apparatus according to claim 1, wherein said circuit is configured to detect optimizable sequences of instruction codes on-the-fly.

11. The apparatus according to claim 1, wherein said circuit comprises a sequence optimization circuit.

12. The apparatus according to claim 1, wherein said circuit comprises a native instruction sequence generator circuit.

13. The apparatus according to claim 1, wherein said circuit is coupled between a processor and a memory system.

14. The apparatus according to claim 13, wherein said circuit is configured to (i) directly connect said processor and said memory system during a first state of operation and (ii) during a second state of operation, communicate with said processor as though said circuit was the memory system and communicate with said memory system as though said circuit was the processor.

15. The apparatus according to claim 1, wherein said instruction codes of said first instruction set comprise Java bytecodes.

16. The apparatus according to claim 1, wherein said circuit comprises a portion of a Java virtual machine implemented in hardware.

17. An apparatus comprising:

means for translating instruction codes of a first instruction set on-the-fly into addresses into a microcode memory containing sequences of instruction codes of a second instruction set that emulate a functionality of the instruction codes of said first instruction set;

means for receiving said instruction codes of said first instruction set; and

means for presenting said sequences of instruction codes of said second instruction set.

18. A method for on-the-fly translation of instructions of a first instruction set into instructions of a second instruction set comprising the steps of:

(A) receiving an instruction code of said first instruction set;

(B) generating an address into a microcode memory in response to said instruction code of said first instruction set using a hardware translator, wherein said address points to a sequence of instruction codes of said second instruction set that will emulate said instruction code of said first instruction set; and

(C) presenting said sequence of instruction codes of said second instruction set.

19. The method according to claim 18, wherein step B comprises the sub-step of:

selecting said address from a look-up table in response to said instruction code of said first instruction set.

20. The method according to claim 19, wherein step C further comprises the sub-step of:

optimizing said sequence of instruction codes of said second instruction set for a particular processor.

21. The apparatus according to claim 1, wherein said sequences of instruction codes of said second instruction set comprise one or more virtual stack references.

22. The apparatus according to claim 1, wherein said microcode memory further comprises one or more of (i) a size for each sequence of instruction codes of said second instruction set, (ii) a value representing how many bytes an instruction uses from said instruction codes of said first instruction set, and (iii) a stack change variable indicating whether the stack increases or decreases due to said instruction codes of said first instruction set and by how much.